

## APPENDIX D

### SEQUENCING CODE - FORTRAN

```
c Sequencing Code started 11/16/04 modified for loop nesting 12/13/04
c for Lock and Dam study
c Read from Ascii file according to 11/9/04 document describing the records
c Declare variables: if not declared variables beginning with i,j,k,m,n are
c assumed to be integer by default
c November 9, 2004 (revised December 13, 2004)

c Ascii file input records from Simulation program to Sequencing program
c Notes for ascii file needed from simulation routine and fed to sequencing
c program for periods of queues at locks.
c Record A: types- integer, real
c Index number of lock, L, where queue exists (the number 20-25); value of |
c parameter p for sequencing program optimization (default p to the value 1. for
c now).
c Record B: type- integer
c Number of TB's currently in queue upstream of lock L:  $N_U$ 
c Record C: type- integer
c Number of TB's currently in queue downstream of lock L:  $N_D$ 
c Record D: types- integer
c Limit on the number of consecutive upstream lockages (CS) and limit on the
c number of consecutive downstream lockages (CD)
c Record E: types- real
c  $OPP_U$  ,  $OPP_D$  ,  $SAME_U$  ,  $SAME_D$  (These are the number of minutes set up time
c required at lock L:  $OPP_U$ =set up time for an upstream TB passing through lock
c L followed by a downstream TB.  $OPP_D$  similar.
c  $SAME_U$ = set up time required between two upstream TB's passing through lock L
c in sequence.  $SAME_D$  similar.
c Record F: types- real, real, integer, integer
c Average steaming time from lock L to lock L-1=ASU
c Average steaming time from lock L to lock L+1=ASD
c Is queue projected at Lock L-1 at a time ASU hours in the future
c (1=yes, 0=no)?
c Is queue projected at Lock L+1 at a time ASD hours in the future
c (1=yes, 0=no)?
c Record G: types- integer, real, real, real
c Indicator to use weighted lockage times (=1) or (=0) if use lockage times
c only. (use 1 for now)
c Weights assigned to: empty barges, full barges, tow boat
c (For now use .1,1,0 for input for weights): In summary use 1,.1,1,0 for
c entire record.
c Record H: types- integer, integer, integer, real
c For each of  $N_U$  TB's in upstream queue at lock L starting with the vessel
c nearest to lock L:
c Vessel ID, # of full barges, # of empty barges, estimated time for passage
c through lock L (in downstream direction)
c Record I: types- integer, integer, integer, real
c For each of  $N_D$  TB's in downstream queue at lock L starting with the vessel
c nearest to lock L.
c Vessel ID, # of full barges, # of empty barges, estimated time for passage
c through lock L (in upstream direction)
c End of ascii file description. Sample input file is named seq.dat
```

```

c input file is seq.dat (file 3)
c output file is lockageorder.dat (file 7)
c debugging output is in sequences.dat (file 4)

integer ivec(1024,10)
integer wtu(100),wtd(100),ptu(100),ptd(100)
real wgtmu(100),wgtmd(100)
integer lock,nu,nd,cu,cd,qu,qd,ngood
real p,oppu,oppd,samsu,samed,asu,asd
integer vessiu(100),fulbu(100),empbu(100)
integer vessid(100),fulbd(100),empbd(100)
real tlocku(100),tlockd(100)
integer ns(20),ne(20),in(20)
character*1 dir(20),uir(20)
real belt(90),bwobj(90),btobj(90)
character*1 bdir(90,20)
real best(90)
data best /90*999999999/
data dir,uir /40*' '/
data mxbrg,mxbst/ 20,90/
open(unit=3,file='C:\LOCK\seq.dat')
open(unit=4,file='C:\LOCK\sequences.dat')
open(unit=7,file='C:\LOCK\lockageorder.dat')
ifile=7
iout=4
write(iout,933)
933 format(' sequencing code 12-13-04')
write(6,911)
911 format(' begin to read file 3 for sequencing code')

c Read ASCII file
read(3,*) lock,p
read(3,*) nu
read(3,*) nd
nur=nu
ndr=nd
read(3,*) cu,cd
read(3,*) oppu,oppd,sameu,samed
read(3,*) asu,asd,qu,qd
read(3,*) iweit,wgtemp,wgtful,wgttow
write(6,92) lock,nu,nd,cu,cd
92 format(' some input from 3=',5i5)
write(6,93) iweit,wgtemp,wgtful,wgttow
93 format(' iweit=',i2,' weights for empty,full,tow=',3f9.2)
if(nu.gt.0)then
do 1 i=1,nu
read(3,*) vessiu(i),fulbu(i),empbu(i),tlocku(i)
write(iout,988)vessiu(i),fulbu(i),empbu(i),tlocku(i)
1 continue
end if
if(nd.gt.0)then
do 2 i=1,nd
read(3,*) vessid(i),fulbd(i),empbd(i),tlockd(i)
write(iout,989)vessid(i),fulbu(i),empbu(i),tlockd(i)

```

```

988     format(' vessel,fulbu,empbu,tlocku=',i6,i6,i6,f7.2)
989     format(' vessel,fulbd,empbd,tlockd=',3i6,f7.2)
2       continue

       end if
c reduce n=nu+nd to mxbrg for now.
       if(nu+nd.gt.mxbrg)then
         sum=nu+nd
         pup=nu/sum
         pdp=nd/sum
         nu=pup*mxbrg
         nd=pdp*mxbrg
6       if(nu+nd.eq.mxbrg) goto 3
5       if(nu+nd.lt.mxbrg)then
         if(nu.gt.nd)then
           nu=nu+1
           goto 4
         end if
         if(nu.le.nd)then
           nd=nd+1
           goto 4
         end if
       end if
       if(nu+nd.gt.mxbrg)then
         if(nu.gt.nd)then
           nu=nu-1
           goto 4
         end if
         if(nu.le.nd)then
           nd=nd-1
           goto 4
         end if
       end if
4       continue
       if(nu+nd.ne.mxbrg)goto 6
       end if
3       continue
       if(nu+nd.ne.nur+ndr)then
         write(iout,44) nu,nd,nur,ndr
44      format(' number of tows reduced to (nu,nd):',2i3,' from ',2i3,/'
1      ' due to program constraints')
       end if
c Use ordering by lockage time or weighted lockage time
       if(nu.gt.0)then
         do 100 i=1,nu
           wtu(i)=wgtful*fulbu(i)+wgtemp*empbu(i)+wgttow
           if(iweit.eq.0) wtu(i)=1.
100      wgtmu(i)=tlocku(i)/wtu(i)
         end if
         if(nd.gt.0)then
           do 101 i=1,nd
             wtd(i)=wgtful*fulbd(i)+wgtemp*empbd(i)+wgttow
             if(iweit.eq.0) wtd(i)=1.
101      wgtmd(i)=tlockd(i)/wtd(i)
           end if
c now sort each side by wgtmd or wgtmu using pointers
         do 150 i=1,nu

```

```

150     ptu(i)=i
        do 151 i=1,nd
151     ptd(i)=i
        call sortwt(wgtmu,nu,ptu)
        write(iout,999) (wgtmu(ptu(i)),i=1,nu)
        write(iout,990) (ptu(i),i=1,nu)
990     format(' ptu=',10i2)
999     format(' wgtmu sorted with ptu=',10f8.2)
        call sortwt(wgtmd,nd,ptd)
        write(iout,998) (wgtmd(ptd(i)),i=1,nd)
998     format(' wgtmd sorted with ptd=',10f8.2)
        write(iout,991) (ptd(i),i=1,nd)
991     format(' ptd=',10i2)
c If no weights desired change the calc of wtu and wtd in 100 and 101 to
c just be the value 1.
c File input is complete
c Start on complete enumeration
c Create number of nested loops and save some good solutions and values
c For example, keep 3 best for starting on one side and and 3 best
c starting with other side
c among those 6 is the optimal
        nvec=0
        do 211 i1=1,0,-1
            do 212 i2=1,0,-1
                do 213 i3=1,0,-1
                    do 214 i4=1,0,-1
                        do 215 i5=1,0,-1
                            do 216 i6=1,0,-1
                                do 217 i7=1,0,-1
                                    do 218 i8=1,0,-1
                                        do 219 i9=1,0,-1
                                            do 220 i10=1,0,-1
                                                nvec=nvec+1
                                                ivec(nvec,1)=i1
                                                ivec(nvec,2)=i2
                                                ivec(nvec,3)=i3
                                                ivec(nvec,4)=i4
                                                ivec(nvec,5)=i5
                                                ivec(nvec,6)=i6
                                                ivec(nvec,7)=i7
                                                ivec(nvec,8)=i8
                                                ivec(nvec,9)=i9
                                                ivec(nvec,10)=i10
220                continue
219            continue
218        continue
217    continue
216    continue
215    continue
214    continue
213    continue
212    continue
211    continue
c End set up of ivec to avoid more than 10 nested loops.
2112 format(' i= ',i4,3x,10i2)
c2111 continue

```

```

n=nu+nd
nloops=n-1
ncas=2**n
write(iout,88)n,ncas,nu,nd,cu,cd,n,nloops
88  format(' n,ncas=',2i9,' nu,nd,cu,cd=',4i3,' n,nloops=',2i6)
nrec=0
ist=1024
ien=1
if(nloops.le.10)then
ien=(1024-2**nloops)+1
end if
700 write(iout,700) ist,ien
format(' at 41..ist,ien=',2i6)
do 41 k=ist,ien,-1
in(1)=ivec(k,10)
in(2)=ivec(k,9)
in(3)=ivec(k,8)
in(4)=ivec(k,7)
in(5)=ivec(k,6)
in(6)=ivec(k,5)
in(7)=ivec(k,4)
in(8)=ivec(k,3)
in(9)=ivec(k,2)
in(10)=ivec(k,1)
if(nloops.le.10)then
c Small problems and then goto 29
nrec=nrec+1
if(nrec.le.100)write(iout,89)
1 nrec,(in(1),l=1,nloops)
89 format(i9,10x,19i2)
call feas(n,ifdu,ifdd,ifuu,ifud,cu,cd,nd,nu,md,mu,nloops,
1 in,dir,uir)
c Check feasibility for cu,cd and nu,nd using feas routine
c If one or more feasible call evaluation routine and save a number of
c best solutions. Then goto 29
if(ifdu+ifdd+ifuu+ifud.le.0)goto 29
if(ifdu+ifdd.eq.2)then
ngood=ngood+1
c write(iout,75)nrec,ifdu,ifdd,(dir(i),i=1,n),md,mu,ngood
75 format(1x,i9,' ifdu,ifdd=',2i2,3x,8(a1,1x),' md,mu,ngd=',3i5,/)
call evalt(dir,p,nu,nd,cu,cd,oppu,oppd,sameu,samed,asu,asd,qu,
1 qd,vessiu,vessid,fulbu,fulbd,empbu,empbd,tlocku,tlockd,best,
2 belt,bwobj,btobj,bdir,wgtmu,wgtmd,ptu,ptd,mxbst,ngood)
end if
if(ifuu+ifud.eq.2)then
ngood=ngood+1
c write(iout,76)nrec,ifuu,ifud,(uir(i),i=1,n),md,mu,ngood
76 format(1x,i9,' ifuu,ifud=',2i2,3x,8(a1,1x),' md,mu,ngd=',3i5,/)
call evalt(uir,p,nu,nd,cu,cd,oppu,oppd,sameu,samed,asu,asd,qu,
1 qd,vessiu,vessid,fulbu,fulbd,empbu,empbd,tlocku,tlockd,best,
2 belt,bwobj,btobj,bdir,wgtmu,wgtmd,ptu,ptd,mxbst,ngood)
end if
goto 29
end if
if(nloops.gt.10)then
ist1=1024
ien1=1

```

```

end if
if(nloops.lt.20)then
  ien1=(1024-2**(nloops-10))+1
end if
do 411 kk=ist1,ien1,-1
  in(11)=ivec(kk,10)
  in(12)=ivec(kk,9)
  in(13)=ivec(kk,8)
  in(14)=ivec(kk,7)
  in(15)=ivec(kk,6)
  in(16)=ivec(kk,5)
  in(17)=ivec(kk,4)
  in(18)=ivec(kk,3)
  in(19)=ivec(kk,2)
  nrec=nrec+1
  if(nrec.le.100)write(iout,89)
1   nrec,(in(1),l=1,nloops)
  call feas(n,ifdu,ifdd,ifuu,ifud,cu,cd,nd,nu,md,mu,nloops,
1   in,dir,uir)
c Check feasibility for cu,cd and nu,nd using feas routine
c If none goto 29 (or more later)
c If one or more feasible call evaluation routine and save a number of
c best solutions. Then goto 29
  if(ifdu+ifdd+ifuu+ifud.le.0)goto 411
  if(ifdu+ifdd.eq.2)then
    ngood=ngood+1
c   write(iout,75)nrec,ifdu,ifdd,(dir(i),i=1,n),md,mu,ngood
    call evalt(dir,p,nu,nd,cu,cd,oppu,oppd,sameu,samed,asu,asd,qu,
1   qd,vessiu,vessid,fulbu,fulbd,empbu,empbd,tlocku,tlockd,best,
2   belt,bwobj,btobj,bdir,wgtmu,wgtmd,ptu,ptd,mxbst,ngood)
  end if
  if(ifuu+ifud.eq.2)then
    ngood=ngood+1
c   write(iout,76)nrec,ifuu,ifud,(uir(i),i=1,n),md,mu,ngood
    call evalt(uir,p,nu,nd,cu,cd,oppu,oppd,sameu,samed,asu,asd,qu,
1   qd,vessiu,vessid,fulbu,fulbd,empbu,empbd,tlocku,tlockd,best,
2   belt,bwobj,btobj,bdir,wgtmu,wgtmd,ptu,ptd,mxbst,ngood)
  end if
c Instead of sub just code eval stuff here directly   call eval
411  continue
29   continue
41   continue

c Test results printout here
c Write out to file for simulation here
  write(ifile,91)lock,cu,cd,n,nur+ndr
91   format(' Lockage order for lock',i3,' limits on consecutive up
1 and down=',2i3,' number of vessels sequenced=',i3,' out of',i3,
2 ' total vessels. ***** denotes vessel not sequenced')
  write(ifile,98)(bdir(1,k),k=1,n)
98   format(' consecutive lockage directions are ',24a1,/,
1' vessel id in order of lockage are:')
  niu=0
  nid=0
  nlock=0
  do 970 k=1,n
    if(bdir(1,k).eq.'U') then

```

```

        niu=niu+1
        nlock=nlock+1
        write(ifile,971) nlock,niu,vessiu(ptu(niu))
971    format(1x,' overall lockage=',i3,' from upstream=',i5,
1' upstream vessel id=',i11,' from upstream')
        end if
        if(bdir(1,k).eq.'D')then
            nid=nid+1
            nlock=nlock+1
            write(ifile,972) nlock,nid,vessid(ptd(nid))
972    format(1x,' overall lockage=',i3,' from downstream=',i3,
1' downstream vessel id=',i9,' from downstream')
            end if
970    continue
c Add in excess over 20 here
        if(nur+ndr.gt.20)then
            if(nu.lt.nur)then
                do 640 i=nu+1,nur
                    nlock=nlock+1
                    niu=niu+1
                    write(ifile,9711)nlock,niu,vessiu(i)
9711    format(1x,' overall lockage=',i3,' from upstream=',i5,
1' upstream vessel id=',i11,' from upstream*****')
640    continue
                end if
                if(nd.lt.ndr)then
                    do 641 i=nd+1,ndr
                        nlock=nlock+1
                        nid=nid+1
                        write(ifile,9721)nlock,nid,vessid(i)
9721    format(1x,' overall lockage=',i3,' from downstream=',i3,
1' downstream vessel id=',i9,' from downstream*****')
641    continue
                    end if
                end if
                mx=min(ngood,mxbst)
                do 96 i=1,mx
                    if(i.le.10.or.i.ge.80) then
                        write(iout,95)i,best(i),belt(i),bwobj(i),btobj(i)
95    format(' 10 best.i.best,belt,bwobj,btobj=',i3,4f12.2)
                        write(iout,97) (bdir(i,k),k=1,nu+nd)
97    format(' bdir...=',20a1)
                    end if
                continue
96    write(iout,90) nrec,ngood
                write(6,90) nrec,ngood
90    format(' stop sequencing code.nrec,ngood=',2i9)
                stop
            end

        subroutine feas(n,ifdu,ifdd,ifuu,ifud,cu,cd,nd,nu,md,mu,
1 nloops,in,dir,uir)
        integer ifd,ifu,cu,cd,nd,nu,nloops,in(20),n
        integer pd,pu,mxnd,mxnu
        character*1 dir(20),uir(20)
c now first check spacing that cannot be too long-check that
c run of zeros is less than cu and/or less than cd

```

```

c If too long then we are infeasible. If feasible set
c ifd and/or ifu equal to 1
c Assume the up boat is first
c Use a separate case for the down boat first.
c However do this once and then just interchange
c values for mu and md to check the other case.
    nzc=0
    mxnc=0
    do 10 i=1,nloops
    if(in(i).eq.1)goto 9
    nzc=nzc+1
    goto 10
9    if(nzc.gt.mxnc.and.(i.lt.n-cu.or.i.lt.n-cd)) mxnc=nzc
    nzc=0
10   continue
    if(mxnc.lt.cu) ifu=1
    if(mxnc.lt.cd) ifd=1
    if(ifd+ifu.le.0)return
c Otherwise we are OK with cu and cd but need to check nd and nu
c Reset ifu and ifd to 0 again for next feas check
    ifdu=0
    ifdd=0
    ifuu=0
    ifud=0
    md=0
    mu=0
c Assume start with a down boat and just switch at end with comparing
c nd and nu against md and mu
    do 20 i=1,nloops+1
    uir(i)=' '
20   dir(i)=' '
    if(in(1).eq.0)then
    dir(1)='D'
    dir(2)='D'
    end if
    if(in(1).eq.1)then
    dir(1)='D'
    dir(2)='U'
    end if
    do 30 i=2,nloops
    if(in(i).eq.0) dir(i+1)=dir(i)
    if(in(i).eq.1.and.dir(i).eq.'D') dir(i+1)='U'
    if(in(i).eq.1.and.dir(i).eq.'U') dir(i+1)='D'
30   continue
c Add a switch here to say no need to check at end of string
    lnd=0
    lnu=0
    do 32 i=n-cu,n
    if(dir(i).ne.'U') goto 33
32   continue
    lnu=1
33   do 36 i=n-cd,n
    if(dir(i).ne.'D')goto 38
36   continue
    lnd=1
38   continue
    mu=0

```

```

        md=0
        do 35 i=1,n
            if(dir(i).eq.'D') md=md+1
            if(dir(i).eq.'U') mu=mu+1
35         continue
            if(md.ne.nd.or.mu.ne.nu)goto 500
c Check again for max consecutive down
c Check consecutive runs down
        istd=0
        iend=0
        ii=0
        mxpd=0
        do 350 i=1,n
            if(i.le.ii)goto 350
            if(dir(i).ne.'D')goto 350
            istd=i
            if(istd.ge.n)goto 3501
            do 3500 ii=istd+1,n
                if(dir(ii).ne.'D')then
                    iend=ii-1
                    goto 349
                end if
3500         continue
3501         iend=n
            goto 3502
349         if(iend.ne.n.and.iend-istd+1.gt.mxpd)mxpd=iend-istd+1
350         continue
3502         continue
            if(mxpd.le.cd)ifdd=1
c Check up consecutives
c Check consecutive runs up
        istu=0
        ienu=0
        ii=0
        mxpu=0
        do 360 i=1,n
            if(i.le.ii)goto 360
            if(dir(i).ne.'U')goto 360
            istu=i
            if(istu.ge.n)goto 3601
            do 3600 ii=istu+1,n
                if(dir(ii).ne.'U')then
                    ienu=ii-1
                    goto 369
                end if
3600         continue
3601         ienu=n
            goto 3602
369         if(ienu.ne.n.and.ienu-istu+1.gt.mxpu)mxpu=ienu-istu+1
360         continue
3602         continue
            if(mxpu.le.cu)ifdu=1
c Check with U starting and see
500         continue
            if(md.eq.nu.and.mu.eq.nd)then
                do 37 i=1,n
                    if(dir(i).eq.'D') then

```

```

        uir(i)='U'
    end if
    if(dir(i).eq.'U') then
        uir(i)='D'
    end if
37      continue
c Check consecutive runs
    istu=0
    ienu=0
    ii=0
    mxpu=0
    do 460 i=1,n
    if(i.le.ii)goto 460
    if(uir(i).ne.'U')goto 460
    istu=i
    if(istu.ge.n)goto 4601
    do 4600 ii=istu+1,n
    if(uir(ii).ne.'U')then
        ienu=ii-1
        goto 4691
    end if
4600  continue
4601      ienu=n
        goto 4602
4691  if(ienu.ne.n.and.ienu-istu+1.gt.mxpu)mxpu=ienu-istu+1
460   continue
4602  continue
        if(mxpu.le.cu)ifuu=1
c Check the 'D' case after the switch
c Check consecutive runs
    istd=0
    iend=0
    ii=0
    mxpd=0
    do 450 i=1,n
    if(i.le.ii)goto 450
    if(uir(i).ne.'D')goto 450
    istd=i
    if(istd.ge.n)goto 4501
    do 4500 ii=istd+1,n
    if(uir(ii).ne.'D')then
        iend=ii-1
        goto 3491
    end if
4500  continue
4501      iend=n
        goto 4502
3491  if(iend.ne.n.and.iend-istd+1.gt.mxp)mxpd=iend-istd+1
450   continue
4502  continue
        if(mxpd.le.cd)ifud=1
    end if
        return
    end

```

Subroutine Evalt(dir,p,nu,nd,cu,cd,oppu,oppd,sameu,samed,asu,asd,  
lqu,qd,vessiu,vessid,fulbu,fulbd,empbu,empbd,tlocku,tlockd,best,

```

2belt,bwobj,btobj,bdir,wgtmu,wgtmd,ptu,ptd,mxbst,ngood)
c Grab declarations from main and add some.
integer ptu(100),ptd(100)
real wgtmu(100),wgtmd(100)
integer lock,nu,nd,cu,cd,qu,qd,ngood
real p,oppu,oppd,samsu,samed,asu,asd
integer vessiu(100),fulbu(100),empbu(100)
integer vessid(100),fulbd(100),empbd(100)
real tlocku(100),tlockd(100)
integer ns(20),ne(20),in(20)
character*1 dir(20),uir(20)
real best(mxbst)
real belt(mxbst),bwobj(mxbst),btobj(mxbst)
character*1 bdir(mxbst,20)
c write(4,50) (dir(i),i=1,20),ngood
50 format(' evalt..dir= ',20a1,' ngood=',i10)
c Skip eval if first direction has a queue
if(qu+qd.eq.1.and.((qu.eq.1.and.dir(1).eq.'U').or.
1 (qd.eq.1.and.dir(1).eq.'D'))) return
n=nu+nd
eltime=0
wobj=0
tobj=0
pobj=0
ncd=0
ncu=0
if(dir(1).eq.'D') then
c First one from the downstream side
wobj=wobj+wgtmd(ptd(1))*n
tobj=tobj+tlockd(ptd(1))*n
eltime=eltime+tlockd(ptd(1))
ncd=ncd+1
end if
if(dir(1).eq.'U') then
wobj=wobj+wgtmu(ptu(1))*n
tobj=tobj+tlocku(ptu(1))*n
eltime=eltime+tlocku(ptu(1))
ncu=ncu+1
end if
c Now look at rest of tb's in sequence dir
do 10 i=2,n
if(dir(i-1).eq.dir(i))then
if(dir(i).eq.'D')then
ncd=ncd+1
eltime=eltime+samed+tlockd(ptd(ncd))
wobj=wobj+wgtmd(ptd(ncd))*(n-ncd-ncu+1)
tobj=tobj+(tlockd(ptd(ncd))+samed)*(n-ncd-ncu+1)
end if
if(dir(i).eq.'U')then
ncu=ncu+1
eltime=eltime+sameu+tlocku(ptu(ncu))
wobj=wobj+wgtmu(ptu(ncu))*(n-ncd-ncu+1)
tobj=tobj+(tlocku(ptu(ncu))+sameu)*(n-ncd-ncu+1)
end if
end if
if(dir(i-1).ne.dir(i))then
if(dir(i).eq.'D')then

```

```

        ncd=ncd+1
        eltime=eltime+oppd+tlockd(ptd(ncd))
        wobj=wobj+wgtmd(ptd(ncd))*(n-ncd-ncu+1)
        tobj=tobj+(tlockd(ptd(ncd))+oppd)*(n-ncd-ncu+1)
    end if
    if(dir(i).eq.'U')then
        ncu=ncu+1
        eltime=eltime+oppu+tlocku(ptu(ncu))
        wobj=wobj+wgtmu(ptu(ncu))*(n-ncd-ncu+1)
        tobj=tobj+(tlocku(ptu(ncu))+oppu)*(n-ncd-ncu+1)
    end if
end if
10    continue
c Put in list of best ones
    fobj=p*tobj+(1-p)*eltime
    if(ngood.lt.1000)
1    write(4,200)fobj,tobj,wobj,eltime,mxbst,ngood,(dir(i),i=1,20)
200    format(' evalt.f,t,w,el,best=',4f7.1,' mxbst=',i3,' ngood=',i10,
1    ' dir= ',20a1)
    do 20 i=1,mxbst
        if(fobj.lt.best(i))then
            if(i.ge.mxbst)return
            do 22 j=mxbst,i+1,-1
                belt(j)=belt(j-1)
                bwobj(j)=bwobj(j-1)
                btobj(j)=btobj(j-1)
            do 231 k=1,nu+nd
231        bdir(j,k)=bdir(j-1,k)
22        best(j)=best(j-1)
            best(i)=fobj
            do 23 k=1,n
23        bdir(i,k)=dir(k)
            belt(i)=eltime
            bwobj(i)=wobj
            btobj(i)=tobj
            return
        end if
20    continue
    return
end
SUBROUTINE SORTWT(BB,N,PT)
c For lock and dam 11/24/04 want increasing order (smallest first)
    INTEGER PT(100)
    DIMENSION BB(100)
    I=1
10    M=I+I+1
        I=I+1
        IF(N.GE.I)GOTO 10
14    M=M/2
        IF(M.EQ.0)RETURN
18    K=N-M
        J=1
20    IF(J.GT.K)GOTO 14
        I=J
24    IF(I.GE.1)GOTO 28
26    J=J+1
        GOTO 20

```

SOR00010

SOR00020

SOR00030

SOR00040

SOR00050

SOR00060

SOR00070

SOR00080

SOR00090

SOR00100

SOR00110

SOR00120

SOR00130

SOR00140

SOR00150

SOR00160

28	MM=I+M	SOR00170
	IF(BB(PT(I))-BB(PT(MM)))26,26,30	SOR00180
30	IT=PT(I)	SOR00190
	PT(I)=PT(MM)	SOR00200
	PT(MM)=IT	SOR00210
	I=I-M	SOR00220
	GOTO 24	SOR00230
	END	SOR00240